In [3]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2018.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27234771 entries, 0 to 27234770
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 2.0+ GB
```

Out[3]:
```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2018        1            14100          12892                0        AA
 1  2018        1            12892          14100                1        99
 2  2018        1            14100          12892                0        AA
 3  2018        1            12892          14100                0        AA
 4  2018        1            14100          12892                0        AA

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1   672.87         2402
 1         0           1   438.13         3099
 2         0           1   367.68         2402
 3         0           1   422.32         2759
 4         0           1   417.94         2402  )
```

In [4]:
```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2018.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]
```

```
# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2018_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2018_f
iltered.csv
```

In [14]:

```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2018_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2018_full_summary_by_ca
```

```
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2018_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [8]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2009.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20421316 entries, 0 to 20421315
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.5+ GB
```

Out[8]:
```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2009        1            11697          14107                0        US
 1  2009        1            14107          11697                0        US
 2  2009        1            11697          14107                0        US
 3  2009        1            14107          11697                0        US
 4  2009        1            11697          14107                0        US

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1   127.00         1972
 1         0           1   127.00         1972
 2         0          17   129.56         1972
 3         0          17   129.56         1972
 4         0           1   130.94         1972  )
```

In [28]:
```python
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2009.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2009_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2009_f
iltered.csv
```

In [15]:
```python
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2009_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
```

```
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2009_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2009_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [16]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2005.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20322876 entries, 0 to 20322875
Data columns (total 10 columns):
 #   Column           Dtype  
---  ------           -----  
 0   Year             int64  
 1   Quarter          int64  
 2   OriginAirportID  int64  
 3   DestAirportID    int64  
 4   TkCarrierChange  int64  
 5   TkCarrier        object 
 6   BulkFare         int64  
 7   Passengers       int64  
 8   MktFare          float64
 9   MktDistance      int64  
dtypes: float64(1), int64(8), object(1)
memory usage: 1.5+ GB
```

Out[16]:

```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2005        1            15096          11292                0        UA
 1  2005        1            11292          15096                0        UA
 2  2005        1            15096          11292                0        UA
 3  2005        1            11292          15096                0        UA
 4  2005        1            15096          10372                0        UA

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1   280.25         1745
 1         0           1   280.73         1748
 2         0           1     5.06         1745
 3         0           1     5.07         1748
 4         0           1   172.04         1620  )
```

In [17]:

```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2005.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2005_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2005_f
iltered.csv
```

In [18]:

```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2005_filtered.csv'
```

```
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2005_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2005_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [19]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2006.csv'
```

```
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20534781 entries, 0 to 20534780
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.5+ GB
```

Out[19]:

(None,

	Year	Quarter	OriginAirportID	DestAirportID	TkCarrierChange	TkCarrier	BulkFare	Passengers	MktFare	MktDistance
0	2006	1	14321	14570	1	99	0	1	224.53	2886
1	2006	1	14570	14321	0	HP	0	1	236.59	3041
2	2006	1	14321	14679	0	US	0	1	314.61	2799
3	2006	1	14679	14321	0	US	0	1	307.30	2734
4	2006	1	14321	14771	0	US	0	1	1220.12	2955

)

In [20]:

```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2006.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2006_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2006_f
iltered.csv
```

In [21]:
```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2006_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2006_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2006_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [22]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2007.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20778751 entries, 0 to 20778750
Data columns (total 10 columns):
 #   Column           Dtype  
---  ------           -----  
 0   Year             int64  
 1   Quarter          int64  
 2   OriginAirportID  int64  
 3   DestAirportID    int64  
 4   TkCarrierChange  int64  
 5   TkCarrier        object 
 6   BulkFare         int64  
 7   Passengers       int64  
 8   MktFare          float64
 9   MktDistance      int64  
dtypes: float64(1), int64(8), object(1)
memory usage: 1.5+ GB
```

Out[22]:
```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2007        1            10257          14100                0        US
 1  2007        1            14100          14321                0        US
 2  2007        1            14321          10397                0        US
 3  2007        1            14321          10397                0        US
 4  2007        1            14321          10397                0        US

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1    82.30          212
 1         0           1   141.69          365
 2         0           1   427.04         1030
 3         0           1   297.98         1030
 4         0           1   316.93         1030  )
```

In [23]:
```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2007.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]
```

```
# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2007_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2007_f
iltered.csv
```

In [24]:

```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2007_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
```

```
output_path = '/Users/alicia/Desktop/899/final project/2007_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2007_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [25]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2008.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

Out[25]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21232963 entries, 0 to 21232962
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.6+ GB
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2008        1            14679          14492                0        US
 1  2008        1            14492          14679                0        US
 2  2008        1            14679          14492                0        US
 3  2008        1            14492          14679                0        US
 4  2008        1            14679          14492                0        US

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1   213.86         2207
 1         0           1   131.19         2705
 2         0           1   107.04         2207
 3         0           1   149.32         2705
 4         0           1   121.83         2207  )
```

In [26]:
```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2008.csv'
data = pd.read_csv(file_path)
```

```
# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2008_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2008_f
iltered.csv
```

In [27]:

```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2008_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)
```

```
# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2008_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2008_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [29]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2010.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22038685 entries, 0 to 22038684
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.6+ GB
```

Out[29]:

(None,

	Year	Quarter	OriginAirportID	DestAirportID	TkCarrierChange	TkCarrier	BulkFare	Passengers	MktFare	MktDistance
0	2010	1	12953	15016	0	US	0	1	182.48	888
1	2010	1	15070	10397	0	US	0	1	249.00	793
2	2010	1	15070	10397	0	US	0	1	306.97	793
3	2010	1	15070	10397	0	US	0	1	1015.99	793
4	2010	1	15070	10397	0	US	0	1	124.98	793

)

In [30]:
```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2010.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2010_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2010_f
iltered.csv
```

In [31]:
```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2010_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'
```

```
# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2010_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2010_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [32]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2011.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22999175 entries, 0 to 22999174
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.7+ GB
```

Out[32]:
```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2011        1            10140          10423                0        WN
 1  2011        1            10140          10423                0        WN
 2  2011        1            10140          10423                0        WN
 3  2011        1            10140          10423                0        WN
 4  2011        1            10140          10423                0        WN

    BulkFare  Passengers  MktFare  MktDistance
 0         0          14     1.98          619
 1         0           2    71.99          619
 2         0           2    90.99          619
 3         0           1    92.97          619
 4         0           3   105.97          619  )
```

In [33]:
```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2011.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2011_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2011_f
iltered.csv
```

In [34]:
```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2011_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
```

```
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2011_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2011_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [35]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2012.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22703842 entries, 0 to 22703841
Data columns (total 10 columns):
 #   Column           Dtype  
---  ------           -----  
 0   Year             int64  
 1   Quarter          int64  
 2   OriginAirportID  int64  
 3   DestAirportID    int64  
 4   TkCarrierChange  int64  
 5   TkCarrier        object 
 6   BulkFare         int64  
 7   Passengers       int64  
 8   MktFare          float64
 9   MktDistance      int64  
dtypes: float64(1), int64(8), object(1)
memory usage: 1.7+ GB
```

Out[35]:

```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2012        1            11292          12173                0        UA
 1  2012        1            12173          11292                1        99
 2  2012        1            11292          12173                0        UA
 3  2012        1            12173          11292                1        99
 4  2012        1            11292          12173                0        UA

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1   341.91         3366
 1         0           1   340.09         3449
 2         0           1   447.98         3366
 3         0           1   459.02         3449
 4         0           1   447.98         3366  )
```

In [36]:

```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2012.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2012_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2012_f
iltered.csv
```

In [37]:

```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2012_filtered.csv'
```

```
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2012_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2012_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [38]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2013.csv'
```

```
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22847363 entries, 0 to 22847362
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.7+ GB
```

Out[38]:

```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2013        1            15016          13303                0        AA
 1  2013        1            13303          11995                0        AA
 2  2013        1            11995          15016                0        AA
 3  2013        1            15016          13303                0        AA
 4  2013        1            13303          12264                0        AA

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1     3.21         1068
 1         0           1     2.13          710
 2         0           1     4.66         1549
 3         0           1   182.75         1068
 4         0           1   157.59          921  )
```

In [39]:

```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2013.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2013_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2013_f
iltered.csv
```

In [40]:
```python
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2013_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2013_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2013_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [41]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2014.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24372712 entries, 0 to 24372711
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.8+ GB
```

Out[41]:

```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2014        1            11618          13930                0        UA
 1  2014        1            13930          11618                0        UA
 2  2014        1            11618          13930                0        UA
 3  2014        1            13930          11618                0        UA
 4  2014        1            11618          13930                0        UA

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1    266.5          719
 1         0           1    266.5          719
 2         0           1    266.5          719
 3         0           1    268.0          719
 4         0           1    268.0          719  )
```

In [42]:
```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2014.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]
```

```
# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2014_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2014_f
iltered.csv
```

In [43]:

```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2014_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
```

```
output_path = '/Users/alicia/Desktop/899/final project/2014_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2014_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [44]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2015.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

Out[44]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25023611 entries, 0 to 25023610
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.9+ GB
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2015        1            12339          14635                0        WN
 1  2015        1            14635          12339                0        WN
 2  2015        1            12339          14635                0        WN
 3  2015        1            14635          12339                0        WN
 4  2015        1            12339          14635                0        WN

    BulkFare  Passengers  MktFare  MktDistance
 0         0           8    255.0          945
 1         0           8    255.0          945
 2         0           3    256.0          945
 3         0           3    256.0          945
 4         0           1    256.5          945  )
```

In [45]:
```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2015.csv'
data = pd.read_csv(file_path)
```

```
# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2015_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2015_f
iltered.csv
```

In [46]:

```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2015_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)
```

```
# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2015_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2015_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [47]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2016.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25535793 entries, 0 to 25535792
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.9+ GB
```

Out[47]:
```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2016        1            14107          14574                0        AA
 1  2016        1            14574          14107                0        AA
 2  2016        1            14107          14574                0        AA
 3  2016        1            14574          11298                0        AA
 4  2016        1            14574          14107                0        AA

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1   458.50         1928
 1         0           1   461.50         1928
 2         0           1   461.50         1928
 3         0           1  1708.00         2796
 4         0           1   245.03         1928  )
```

In [48]:
```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2016.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2016_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2016_f
iltered.csv
```

In [49]:
```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2016_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
```

```
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2016_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2016_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [50]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2017.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25615030 entries, 0 to 25615029
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.9+ GB
```

Out[50]:

```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2017        1            14107          13891                0        AA
 1  2017        1            14107          13891                0        AA
 2  2017        1            14107          13891                0        AA
 3  2017        1            14107          13891                0        AA
 4  2017        1            14107          13891                0        AA

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1    136.0          325
 1         0           2    153.0          325
 2         0           1    155.0          325
 3         0           1    156.0          325
 4         0           8    163.0          325  )
```

In [51]:

```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2017.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2017_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2017_f
iltered.csv
```

In [52]:

```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2017_filtered.csv'
```

```
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2017_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2017_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [53]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2018.csv'
```

```
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27234771 entries, 0 to 27234770
Data columns (total 10 columns):
 #   Column           Dtype
---  ------           -----
 0   Year             int64
 1   Quarter          int64
 2   OriginAirportID  int64
 3   DestAirportID    int64
 4   TkCarrierChange  int64
 5   TkCarrier        object
 6   BulkFare         int64
 7   Passengers       int64
 8   MktFare          float64
 9   MktDistance      int64
dtypes: float64(1), int64(8), object(1)
memory usage: 2.0+ GB
```

Out[53]:

(None,

	Year	Quarter	OriginAirportID	DestAirportID	TkCarrierChange	TkCarrier	BulkFare	Passengers	MktFare	MktDistance
0	2018	1	14100	12892	0	AA	0	1	672.87	2402
1	2018	1	12892	14100	1	99	0	1	438.13	3099
2	2018	1	14100	12892	0	AA	0	1	367.68	2402
3	2018	1	12892	14100	0	AA	0	1	422.32	2759
4	2018	1	14100	12892	0	AA	0	1	417.94	2402

)

In [54]:

```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2018.csv'
data = pd.read_csv(file_path)

# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2018_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2018_f
iltered.csv
```

In [55]:
```python
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2018_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)

# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2018_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2018_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [1]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/2019.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28535894 entries, 0 to 28535893
Data columns (total 10 columns):
 #   Column           Dtype  
---  ------           -----  
 0   Year             int64  
 1   Quarter          int64  
 2   OriginAirportID  int64  
 3   DestAirportID    int64  
 4   TkCarrierChange  int64  
 5   TkCarrier        object 
 6   BulkFare         int64  
 7   Passengers       int64  
 8   MktFare          float64
 9   MktDistance      int64  
dtypes: float64(1), int64(8), object(1)
memory usage: 2.1+ GB
```

Out[1]:
```
(None,
    Year  Quarter  OriginAirportID  DestAirportID  TkCarrierChange TkCarrier
\
 0  2019        1            11057          14524                0        AA
 1  2019        1            14524          11057                0        AA
 2  2019        1            11057          14524                0        AA
 3  2019        1            14524          11057                0        AA
 4  2019        1            11057          14524                0        AA

    BulkFare  Passengers  MktFare  MktDistance
 0         0           1    352.5          257
 1         0           1    352.5          257
 2         0           1    353.0          257
 3         0           1    353.0          257
 4         0           1    353.5          257  )
```

In [2]:
```
import pandas as pd

# Read the dataset
file_path = '/Users/alicia/Desktop/899/final project/2019.csv'
data = pd.read_csv(file_path)
```

```
# Apply filters to remove unwanted rows
filtered_data = data[
    (data['TkCarrierChange'] != 1) &
    (data['TkCarrier'] != 99) &
    (data['MktFare'] >= 25) &
    (data['MktFare'] <= 2500)
]

# Save the filtered dataset to a new CSV file
output_path = '/Users/alicia/Desktop/899/final project/2019_filtered.csv'
filtered_data.to_csv(output_path, index=False)

print(f"Filtered data has been saved to {output_path}")
```

```
Filtered data has been saved to /Users/alicia/Desktop/899/final project/2019_f
iltered.csv
```

In [3]:

```
import pandas as pd

# Load the filtered dataset
file_path = '/Users/alicia/Desktop/899/final project/2019_filtered.csv'
data = pd.read_csv(file_path)

# Group by `TkCarrier` and `Quarter` to calculate summary metrics
carrier_summary = (
    data.groupby(['TkCarrier', 'Quarter'])
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Calculate harmonic mean cost per mile for each carrier and quarter
carrier_summary['HarmonicCostPerMile'] = (
    carrier_summary['Passengers'] /
    (carrier_summary['MktDistance'] * carrier_summary['Passengers'] / carrier_s
)

# Calculate quarterly market summary
market_summary = (
    data.groupby('Quarter')
    .agg({
        'Passengers': 'sum',              # Total passengers
        'MktFare': 'mean',                # Average market fare
        'MktDistance': 'mean'             # Average market distance
    })
    .reset_index()
)

# Add a placeholder for `TkCarrier` in the market summary
market_summary['TkCarrier'] = 'Total Market'

# Calculate harmonic mean cost per mile for the entire market
market_summary['HarmonicCostPerMile'] = (
    market_summary['Passengers'] /
    (market_summary['MktDistance'] * market_summary['Passengers'] / market_summ
)
```

```
# Combine carrier-level and market-level summaries
full_summary = pd.concat([carrier_summary, market_summary], ignore_index=True)

# Sort by `TkCarrier` and `Quarter`
full_summary = full_summary.sort_values(by=['TkCarrier', 'Quarter'])

# Save the full summary table to a CSV file
output_path = '/Users/alicia/Desktop/899/final project/2019_full_summary_by_ca
full_summary.to_csv(output_path, index=False)

print(f"Full summary table saved to {output_path}")
```

```
Full summary table saved to /Users/alicia/Desktop/899/final project/2019_full_
summary_by_carrier_and_quarter.csv
```

In [ ]:

In [1]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/airport_market_data.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   origin_airport_id  244 non-null    int64 
 1   origin_city        244 non-null    object
 2   population         244 non-null    int64 
 3   income             244 non-null    int64 
dtypes: int64(3), object(1)
memory usage: 7.8+ KB
```

Out[1]:
```
(None,
    origin_airport_id                    origin_city  population  income
 0              10135  Allentown/Bethlehem/Easton, PA      833049   35310
 1              10140                 Albuquerque, NM      905174   33390
 2              10158              Atlantic City, NJ      273035   33530
 3              10170                    Kodiak, AK       99639   47670
 4              10208                    Augusta, GA      590047   31450)
```

In [7]:
```
import pandas as pd
import numpy as np

file_path = '/Users/alicia/Desktop/899/final project/state_summary.csv'
data = pd.read_csv(file_path)

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   State               51 non-null     object 
 1   Average Population  51 non-null     float64
 2   Average Income      51 non-null     float64
dtypes: float64(2), object(1)
memory usage: 1.3+ KB
```

Out[7]:

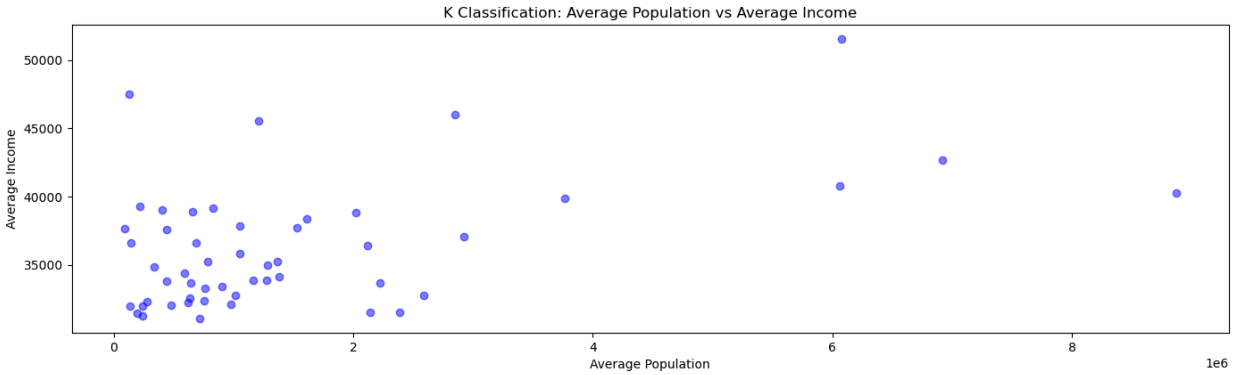
```
(None,
   State  Average Population  Average Income
 0    AK        1.268929e+05    47489.090909
 1    AL        5.944945e+05    34405.000000
 2    AR        6.224955e+05    32235.000000
 3    AZ        2.586906e+06    32742.500000
 4    CA        3.769456e+06    39832.000000)
```

In [8]:

```
import matplotlib.pyplot as plt
import pandas as pd

# Load the data
file_path = '/Users/alicia/Desktop/899/final project/state_summary.csv'
data = pd.read_csv(file_path)

# Create a scatter plot for K-classification based on Population and Income
plt.figure(figsize=(14, 8))

# Plot the scatter for Average Population vs Average Income
plt.subplot(2, 1, 1)
plt.scatter(data['Average Population'], data['Average Income'], c='blue', alpha
plt.title('K Classification: Average Population vs Average Income')
plt.xlabel('Average Population')
plt.ylabel('Average Income')



# Save the combined figure
output_file = '/Users/alicia/Desktop/899/final project/state_classification_and
plt.tight_layout()
plt.savefig(output_file)

output_file
```

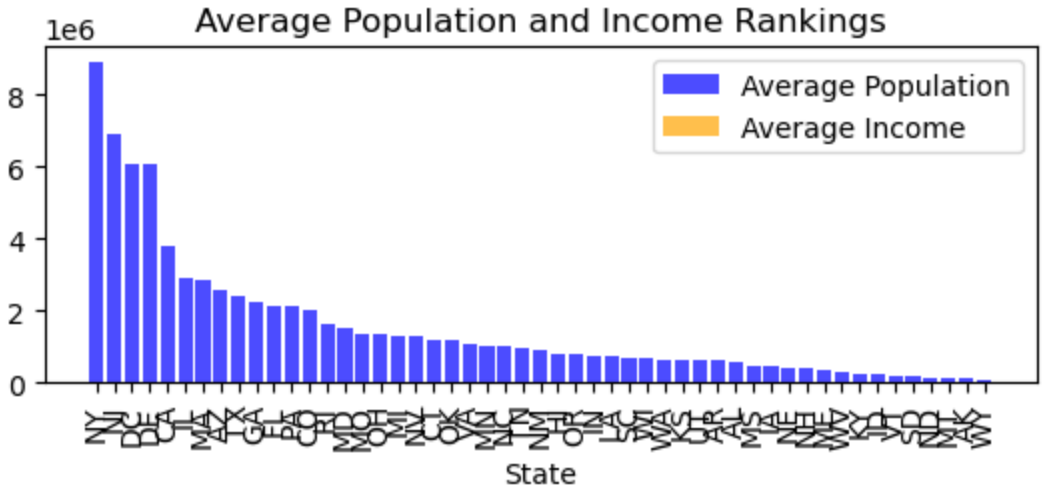
Out[8]:

```
'/Users/alicia/Desktop/899/final project/state_classification_and_ranking.png'
```



In [9]:
```
# Sort data by Average Population and Average Income
data_sorted_population = data.sort_values(by='Average Population', ascending=Fa
data_sorted_income = data.sort_values(by='Average Income', ascending=False)

# Plot the ranking of Average Population
plt.subplot(2, 1, 2)
plt.bar(data_sorted_population['State'], data_sorted_population['Average Popula
plt.bar(data_sorted_income['State'], data_sorted_income['Average Income'], alpl
plt.title('Average Population and Income Rankings')
plt.xlabel('State')
plt.xticks(rotation=90)
plt.legend()
```

Out[9]:
```
<matplotlib.legend.Legend at 0x1675e10d0>
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: